



22883

PATENT TRADEMARK OFFICE

155.1005.01

1 This application is filed in the name of the following inventor(s):

2

3	<i>Inventor Name</i>	<i>Citizenship</i>	<i>Residence City and State</i>
4	Stephane KASRIEL	France	Mountain View, California
5	Walter MANN	United States	San Francisco, California

6

7 The assignee is *Fireclick, Inc.*, a corporation having a place of business at
8 289 South San Antonio Road, Los Altos, California 94022.

9

TITLE OF THE INVENTION

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

1. *Field of the Invention*

17

18 This invention relates to a content delivery network using differential cach-
19 ing.

20

1 2. *Related Art*

2

3 When multiple users (at client devices) request information from a server

4 (at a server device), it often occurs that the number of requests from client devices taxes

5 the server device, and reduces the quality of service that each user experiences. More-

6 over, when those multiple client devices are distributed at widely disparate locations,

7 there is reduced quality of service experienced by users relatively distant from the server

8 device, due to distance (either measured physically or measured by a communication

9 network topology) that messages travel. Accordingly, it would be advantageous to pro-

10 vide additional server devices having the same content for delivery to client devices, to

11 (1) share the load of requests made to the server device, and to (2) move the content for

12 delivery closer to client devices. Each of these effects should improve the quality of ser-

13 vice experienced by client devices.

14

15 One known method is to provide a content delivery network, including an

16 originating server device and a set of mirroring server devices, disposed so that original

17 content from the originating server is delivered and maintained at each of the mirroring

18 servers. While this known method generally achieves the goal of moving content for de-

19 livery closer to client devices, it has the drawback that it is relatively unsuitable for con-

20 tent that is not static. When the content for delivery is dynamically changing, or is per-

21 sonalized for users at different client devices, the content is not static, and the mirroring

22 servers cannot guarantee that they have the correct content for delivery. The content de-

livery network thus is relatively unsuitable for responding to requests for non-static content.

Accordingly, it would be desirable to provide a technique for serving relatively non-static content for delivery in a content delivery network.

SUMMARY OF THE INVENTION

The invention provides a method and system for serving relatively non-static content for delivery in a content delivery network. Content for delivery is pushed out from an originating server to a set of mirroring servers using differential caching. Using differential caching, each object that might be requested by a client is separated into template information and delta information. In a preferred embodiment, the originating server determines a set of templates for differential caching, and causes those templates to be distributed using the content delivery network to the mirroring servers. Each mirroring server in the content delivery network is able to access, either locally or by reference to a nearby distribution point, a copy of an applicable recent template for each request made by a client. Hosting of the template information is decoupled from hosting of the delta information.

Delta information can include (a) changes to the template information since last distributed using the content delivery network, or (b) personalized information spe-

1 cific to the client or to the request made by the client. In a preferred embodiment, delta
2 information can be served separately from the originating server, or can be served sepa-
3 rately from mirroring servers using a content distribution network (possibly the same
4 content distribution network as used to serve template information).

5
6 Using differential caching with the content distribution network (that is, de-
7 coupling distribution of template information from distribution of delta information) al-
8 lows the system to provide the following functions, not provided by known methods:

- 9
10 • The amount of bandwidth used to distribute changes to content from the originat-
11 ing server is minimized. This allows more content to be distributed using the
12 same amount of bandwidth, or alternatively, reduces the cost of distributing the
13 same amount of content.
- 14
15 • The content distribution network is able to distribute dynamically changing con-
16 tent with substantially less use of bandwidth, and with substantially greater re-
17 sponsiveness to the dynamic changes. The template information is sufficiently
18 static for distribution using the content distribution network. Distribution of the
19 delta information does not put an excessive burden on the originating server, or
20 when the delta information is distributed using a content distribution network,
21 does not put an excessive burden on the content distribution network.

- The content distribution network is able to distribute personalized content. As with dynamically changing content, the template information is sufficiently static for distribution using the content distribution network. Distribution of the delta information can be distributed from the same server or a different set of servers. For example, if personalized content requires a database lookup, the originating server or another server can perform primarily database lookups, while a separate mirroring system for the personalized information can distribute the delta information to clients in personalized form.

- There are many individual objects that might be included in a page that is specific to a particular client. While these individual objects are themselves typically very similar or even identical to objects to be included in pages specific to different clients, the selection of which individual objects to be included is responsive to personalized information about each client. Thus, distribution of the individual objects can be decoupled from personalization using those objects, just like distribution of the template information can be decoupled from distribution of delta information.

The invention can be used in conjunction with a known content distribution network, so as to allow the known content distribution network to provide both dynamic content and personalized content (when used in conjunction with the invention).

1 The invention has general applicability to content delivery, not limited spe-
2 cifically to the web pages, web protocols, or caching (and not limited specifically to con-
3 tent delivery as described herein). For example, embodiments of the invention can in-
4 clude one or more of, or some combination of, the following applications:

- 6 • distribution of databases responses, including responses to common or fre-
7 quently used database queries;
- 8 • distribution of email and groupware messages, bulletin board or newsgroup
9 messages; and
- 10 • distribution of streaming media content (template information and insertion
11 points can be streamed from a content delivery network and delta information
12 can be streamed from an originated server

13 Moreover, techniques used by a preferred embodiment of the invention for
14 content delivery can be used in contexts other than the specific applications disclosed
15 herein. For example, techniques used by embodiments of the invention for content deliv-
16 ery are all generally applicable to fields other than the specific applications disclosed
17 herein.
18
19

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows a block diagram of a content delivery network using differential caching.

Figures 2A and 2B show a process flow diagram of a method for operating a content delivery network using a proxy encoder server and one or more decoders.

Figure 3 shows a process flow diagram for a method of operating a content delivery network using differential caching.

DESCRIPTION OF THE PREFERRED EMBODIMENT

The invention is described herein with regard to preferred steps and data structures. Those skilled in the art will recognize, after perusal of this application, that the described steps and data structures are not limited to any particular processing devices (whether general-purpose or special-purpose processing devices, or specific circuitry). Rather, those of ordinary skill in the art would be able to implement the described steps and data structures, and equivalents thereof, without undue experimentation or further invention. All such implementations are within the scope and spirit of the invention.

1 Inventions described herein can be used in conjunction with inventions de-
2 scribed in the following applications:

- 3
- 4 • Application Serial No 09/436/136, filed November 11, 1999, in the name of
5 Stephane KASRIEL, titled "Predictive Pre-Download of Network Objects", attor-
6 ney docket number 155.1001.02;
 - 7
 - 8 • Application Serial No. 09/734,910, filed December 11, 2000, in the name of
9 Stephane KASRIEL, titled "Predictive Pre-Download Using Normalized Network
10 Objects", attorney docket number 155.1002.01;
 - 11
 - 12 • Application Serial No. 09/827,268, filed April 4, 2001, in the name of Stephane
13 KASRIEL, titled "Server-Originated Differential Caching", attorney docket num-
14 ber 155.1003.02;
 - 15

16 Each of these applications is hereby incorporated by reference as if fully set
17 forth herein. They are collectively referred to as the "incorporated disclosures".

1 *Lexicography*

- 2
- 3 • **client** and **server** – as used herein, the phrases, “**client**” and “**server**” refer to a
4 relationship between two devices, particularly to their relationship as client and
5 server, not necessarily to any particular physical devices.
6
 - 7 • **client device** and **server device** – as used herein, the phrase “**client device**” in-
8 cludes any device taking on the role of a client in a client-server relationship (such
9 as an HTTP web client and web server). There is no particular requirement that
10 any client devices must be individual physical devices; they can each be a single
11 device, a set of cooperating devices, a portion of a device, or some combination
12 thereof. As used herein, the phrase “**server device**” includes any device taking on
13 the role of a server in a client-server relationship. There is no particular require-
14 ment that server devices must be individual physical devices; they can each be a
15 single device, a set of cooperating devices, a portion of a device, or some combi-
16 nation thereof.
17
 - 18 • **logically remote** – as used herein, the phrase “**logically remote**” refers to the rela-
19 tive logical placement or degree of connectivity between two or more computer-
20 ized systems or two or more elements within a single system. Generally, elements
21 that are relatively proximate to each other may be logically remote if there is a
22 small probability that information will flow between them on a regular basis.

System Elements

Figure 1 shows a block diagram of a content delivery network using differential caching.

A system 100 includes one or more clients 110, an originating server 120, a set of mirroring servers 130, a set of proxy encoder servers 140 and a communication network 150.

Client

Each client 110 includes a client workstation 111 and a client operator 112.

As described in the incorporated disclosures, a “workstation” might include a personal computer, a software package on a server, a handheld computer cooperating with a personal computer or with a server (or both), or a telephone interface to a system such as an interactive voice response system. There is also no particular requirement that multiple workstations used by a single client need be of the same type. Also as described in the incorporated disclosures, an “operator” might comprise an individual person, a set of persons having authority to act in particular way, a proxy for an individual person or set of persons, such as a human secretary or a computer program having the function of

1 forwarding or aggregating or scheduling requests made by others, or even an artificial in-
2 telligence program such as an expert system or otherwise.

3
4 In a preferred embodiment, each client 110 includes a web browser 113,
5 such as the "Internet Explorer" product or the "Netscape Navigator" product, capable of
6 using a message transfer protocol, such as HTTP (hypertext transfer protocol), or a vari-
7 ant thereof, to request documents (such as for example web pages) from the originating
8 server 120 or a mirroring server 130 and to receive documents and other responses from
9 the originating server 120 or a mirroring server 130. A decoder 114 is coupled to the web
10 browser 113, preferably as a browser add-on. However, in other embodiments, multiple
11 decoders 114 may be situated relatively proximate to an Internet service provider, an en-
12 terprise cache or at other locations within the communications network 150.

13
14 In other embodiments, the browser 113 is not coupled to a decoder 114. In
15 such embodiments, functions normally performed on the client side by the decoder 114
16 are performed by the browser 113 and the proxy encoder server 140. These embodiments
17 are referred to as "clientless versions".

18 19 Originating Server

20
21 The originating server 120 includes a computer 121 and a database 122 of
22 documents 123. In a preferred embodiment, documents 123 can include (as further de-

scribed herein) web pages, embedded objects for web pages, template web pages, changed data for insertion into template web pages, and code fragments.

The originating server 120 includes a processor, program and data memory, and operates under control of software to perform the tasks described herein. In particular, the originating server 120 is capable of using a message transfer protocol, such as HTTP or a variant thereof, to receive requests for documents (such as for example web pages) from clients 110 and to respond to those requests by sending those documents to clients 110. In a preferred embodiment, the originating server 120 uses HTTP version 1.1, or at least some features thereof, as described herein.

Mirroring Servers

Similar to the originating server 120, the mirroring servers 130 each include a computer 131 and a database 132 of documents 133. Similar to the originating server 120, each mirroring server 130 includes a processor, program and data memory, and operates under control of software to perform the tasks described herein.

Proxy encoder servers

Similar to the originating server 120 and the mirroring servers 130, the proxy encoder servers 140 each include a computer 141 and a database 142 of documents

1 133 or template information 124. However, the proxy encoder server 140 also includes a
2 software element 143 that recognizes a URL and transparently alters the URL so as to
3 direct requests from the client 110 to the content delivery network. This software ele-
4 ment 143 can also embed information in the URL such as may be relevant to which ver-
5 sion of a web page or template information is transmitted, whether a decoder 114 can ac-
6 cept delta information 125 and other information that pertains to equipment or communi-
7 cation parameters.

8
9 In a preferred embodiment, the proxy encoder server 140 is relatively local
10 to the originating server 120 and the mirroring servers 130. The encoder 140 may also be
11 integrated inside an existing server component, such as any of the previously described
12 servers, a web server, an application server, a cache or an L7 switch.

13
14 It would be clear to those skilled in the art, after perusal of this application,
15 that the system 100 can include more than one originating server 120, each disposed for
16 originating a set of content for distribution (probably a completely different set of such
17 content for distribution) to clients 110. Moreover, it would be clear to those skilled in the
18 art, after perusal of this application, that at least some of the originating servers 120 can
19 operate as mirroring servers 130 for other, different, originating servers 120, while at
20 least some of the mirroring servers 130 can operate as originating servers 120 for their
21 own specific content for distribution. All such variations and extensions of ideas pre-

1 sented herein are within the scope and spirit of the invention, and would not require either
2 undue experimentation or invention when considered in view of this application.

4 Communication Network

6 Clients 110, originating servers 120 and mirroring servers 130 are coupled
7 using a communication network 150. In a preferred embodiment, the communication
8 network 150 includes a computer communication network, such as the Internet. How-
9 ever, in alternative embodiments, the communication network 150 might include an
10 intranet, extranet, VPN (virtual private network), ATM system, a portion of a private or
11 public PSTN (public switched telephone network), a frame relay system, or any other
12 communication technique capable of performing the functions described herein.

14 Content Distribution

16 In a preferred embodiment, each mirroring server 130 performs a function
17 much like a caching device. Each mirroring server 130 receives requests from clients
18 110, and determines whether (a) it can service those requests directly, or (b) it can service
19 those requests by obtaining information from the originating server 120. When the mir-
20 roring server 130 can service those requests directly, it does so, sending content for deliv-
21 ery to the requesting client 110 in response to the request. When the mirroring server 130
22 can service those requests by obtaining information from the originating server 120, it

1 does so, requesting the content from the originating server 120, and sending that content
2 to the requesting client 110. The mirroring server 130 can determine whether to cache
3 (or otherwise maintain) a copy of the content for delivery, so that later requesting clients
4 110 can be serviced without resort to a request to the originating server 120.

5
6 When maintaining a document 123 (such as a web page), the originating
7 server 120 determines a set of template information 124 and a set of delta information
8 125. In a preferred embodiment, the template information 124 indicates those portions of
9 the document 123 that are relatively static, while the delta information 125 indicates
10 those portions of the document 123 that are relatively dynamic.

11
12 In a preferred embodiment, requests are made by the client 110 using the
13 decoder 114. The decoder 114 forwards the request to the proxy encoder server 140,
14 which in turn, transmits the request to the originating server 120. The original HTTP re-
15 sponse is preferably altered so that additional information can be appended to it, such as
16 whether template information 124 or an entire page was fetched, what version of the
17 template information 124 was obtained and similar information pertaining to other com-
18 munication parameters. This additional information can be used in subsequent requests,
19 so as to determine the relative freshness of a page. The proxy encoder server 140 re-
20 trieves the document 123 and uses it to create an associated template 124. In this em-
21 bodiment, the document 123 is integrated at either the client device 110 by the client 110

1 or at the proxy encoder server 140 (if the client device 110 does not accept delta informa-
2 tion 125).

3
4 In another preferred embodiment, a request from a client 110 can be associ-
5 ated with delta information 125. The delta information 125 is associated with a pointer to
6 a template 124 that is available, either through the content delivery network (if the tem-
7 plate is available there) or at the originating server 120.

8
9 In an alternative embodiment, each mirroring server 130 maintains a copy
10 of template information 124 for that particular document 123. When the document 123 is
11 requested by a client 110, the mirroring server 130 can provide the template information
12 124 to the client 110 from its cache, while the client 110 obtains the delta information
13 125 from the originating server 120 (or from a content distribution network similarly dis-
14 posed for distributing delta information 125). Sending the template information 124 from
15 the originating server 120 to the mirroring servers 130 (and from the mirroring servers
16 130 to the decoder 114) is separate from sending the delta information 125 from the
17 originating server 120 to the client 110.

18 19 *Method of Operation*

20
21 Figures 2A and 2B show a process flow diagram for a method of operating
22 a content delivery network using a proxy encoder server and one or more decoders.

1 A method 200 includes a set of flow points and process steps as described
2 herein.

3
4 Although by the nature of textual description, the flow points and process
5 steps are described sequentially, there is no particular requirement that the flow points or
6 process steps must be sequential. Rather, in various embodiments of the invention, the
7 described flow points and process steps can be performed in a parallel or pipelined man-
8 ner, either by one device performing multitasking or multithreading, or by a plurality of
9 devices operating in a cooperative manner. Parallel and pipelined operations are known
10 in the art of computer science.

11
12 At a flow point 210, a client 110 is ready to make a request for a document
13 123 from a mirroring server 130 or an originating server 120 (depending upon which is
14 closest). In a preferred embodiment, each request for a document 123 is performed inde-
15 pendently, even if a plurality of requests are to be performed substantially simultane-
16 ously.

17
18 At a step 211, the client 110 generates a request message 151 to the mirror-
19 ing server 130 or the originating server 120 for the document 123. The request message
20 151 identifies the document 123 and requests that the mirroring server 130 or the origi-
21 nating server 120 send the document 123 to the client 110. In a preferred embodiment,

1 the request message 151 is made using the decoder 114, preferably on the client's web
2 browser 113.

3
4 In a step 212, the decoder 114 forwards this request message 151 to the
5 proxy encoder 140, situated preferably near the mirroring server 130 or the originating
6 server 120. In forwarding this request, the decoder 114 indicates that the sender, (that is
7 the client 110) is compatible with systems that provide delta encoding.

8
9 In clientless versions (that is, those clients without a decoder 114) the re-
10 quest goes directly to the encoder 140.

11
12 In a step 213, the proxy encoder server 140 fetches document 123 or a tem-
13 plate 124 that corresponds to document 123 from either the originating server 120 or the
14 mirroring server 130, depending upon which is closest to the proxy encoder 140. Upon
15 obtaining the content, the proxy encoder 140 updates the template 124 for the document
16 123. In the event that there is not a template 124 associated with document 123, the
17 proxy encoder 140 generates a template 124 and caches it.

18
19 In clientless embodiments, the proxy encoder 140 responds directly with
20 the delta information. In such embodiments, the delta is an HTML page that includes a
21 reference to a template (for example, a Javascript) and a delta (for example, some
22 Javascript instructions). In this embodiment, the Javascript instructions comprising the

1 delta tell the browser how to transform the template into the correct HTML or XML
2 document.

3
4 In a step 214, the decoder 114 requests the template 124 from either the
5 proxy encoder server 140 or the mirroring server 130, (depending where the encoder
6 proxy 140 specified the template 124 was cached in the previous step, either immediately
7 or at a later point in time, depending upon preferences set by the client operator 112.

8
9 In the clientless version, the browser 113 automatically and immediately re-
10 trieves the template from the site specified in the previous step.

11
12 In a step 215, the encoder proxy 140 sends the contents of the template 124,
13 and a tag (also known as an "Etag") that corresponds to the version of the template. The
14 encoder marks the template as being cacheable by network elements such as the mirror-
15 ing servers 130, the client 110, the decoder 114 or public caches such as HTTP proxy
16 caches.

17
18 If the contents and etag are sent from the mirroring server 130, then the
19 mirroring server 130 searches its cache for the template 124. If the template 124 is pre-
20 sent in its cache, the mirroring server 130 sends the template 124 directly to the client
21 110. If the template 124 is not present in the cache, the mirroring server 130 automati-

1 cally fetches the template 124 from the encoder proxy 140, caches the fetched template
2 124, and sends the template to the client 110.

3
4 The following steps occur when the user subsequently requests the docu-
5 ment 123.

6
7 In a step 216, the same or a different client 110 requests a document 123 by
8 generating a request message 151. Similar to step 210 and 211, this is mediated by the
9 decoder 114 so that the request is directed to the proxy encoder 140. Continuing with the
10 example, the decoder 114 changes the original request to as to further specify a version
11 number that is used to ascertain if changes have occurred.

12
13 In a step 217, the proxy encoder 140 receives the request from the decoder
14 114. If the proxy encoder 140 determines that there is not a version of the template 124
15 in the database 142, the proxy encoder 140 obtains the document 123 from either the
16 originating server 120 or the mirroring server 130 as in steps 213 – 215 and caches the
17 template. If, however, the template 124 is available, the proxy encoder 140 calculates the
18 differences between the version of template 124 that is available in the database 142 and
19 a newer version of the document 123 such as may be available from the originating
20 server 120 or the mirroring server 130. These differences are the delta information 125.

1 In a step 218, the proxy encoder 140 may send either the delta information
2 125 to the decoder 114 (that is, if the decoder 114 can accept delta information 125) or it
3 sends the document 123 to the client.
4

5 Steps 216 through 218 are described with respect to a client – server im-
6 plementation. In the clientless version, the proxy encoder server 140 does not need to
7 know which version of the template 124 is at the client 110. This information is not
8 needed because the proxy encoder server 140 makes this decision a priori and instructs
9 the client 110 to use a specific version of the template 124. Under these circumstances,
10 steps 216 – 218 in the clientless version are comparable to step 210 through 215 in
11 which the browser 113 (rather than the decoder 114) makes a request that is directed to
12 the encoder proxy 140.
13

14 Figure 3 shows a process flow diagram for a method of operating a content
15 delivery network using differential caching.
16

17 A method 300 includes a set of flow points and process steps as described
18 herein.
19

20 Similar to method 200, the flow points and process steps are described se-
21 quentially, there is no particular requirement that the flow points or process steps must be
22 sequential. Rather, in various embodiments of the invention, the described flow points

1 and process steps can be performed in a parallel or pipelined manner, either by one de-
2 vice performing multitasking or multithreading, or by a plurality of devices operating in a
3 cooperative manner. Parallel and pipelined operations are known in the art of computer
4 science.

5 At a flow point 310, a client 110 is ready to make a request for a document
6 123 from a mirroring server 130. In a preferred embodiment, each request for a docu-
7 ment 123 is performed independently, even if a plurality of requests are to be performed
8 substantially simultaneously.

9
10 At a step 311, the client 110 generates a request message 151 (shown in
11 figure 1) to the mirroring server 130 for the document 123. The request message 151
12 identifies the document 123 and requests that the mirroring server 130 send the document
13 123 to the client 110.

14
15 At a step 312, the mirroring server 130 determines if it has template infor-
16 mation 124 for the requested document 123. To perform this step, the mirroring server
17 130 performs one the following sub-steps:

- 18
- 19 • At a sub-step 312(a), if the mirroring server 130 does not have the template infor-
20 mation 124, it generates a request message 152 (shown in figure 1) to the originat-
21 ing server 120 for the template information 124. As part of this sub-step, upon re-

1 ceiving the template information 124, the mirroring server 130 records the tem-
2 plate information 124 in a cache relatively local to the mirroring server 130.

- 3
- 4 • At a sub-step 312(b), the mirroring server 130 has the template information 124,
5 the method proceeds with the next step.
- 6

7 At a step 313, the mirroring server 130 generates a response message 153
8 (shown in figure 1) to the client 110 with the template information 124.

9

10 At a step 314, the client 110 (or the mirroring server 130, if the system 100
11 is so configured) generates a delta request message 154 to the originating server 120 for
12 the delta information 125. As part of this step, the system 100 performs one of the fol-
13 lowing sub-steps:

- 14
- 15 • At a sub-step 314(a), if the client 110 generates the delta request message 154, the
16 client 110 receives the delta information message 155 from the originating server
17 120
 - 18
 - 19 • At a sub-step 314(b), if the mirroring server 130 generates the delta request mes-
20 sage 154, the mirroring server 130 receives the delta information message 155
21 from the originating server 120. As part of this sub-step, the mirroring server 130

1 forwards the delta information 125 in a second delta information message 155 to
2 the client 110.

3
4 At a step 315, the client 110 integrates the template information 124 and the
5 delta information 125 into a unified document 123 for presentation. Integration of the
6 template information 124 and the delta information 125 is further described in the incor-
7 porated disclosures.

8 9 *Generality of the Invention*

10
11 The invention has general applicability to content delivery, not limited spe-
12 cifically to the web pages, web protocols, or caching (and not limited specifically to con-
13 tent delivery as described herein). For example, embodiments of the invention can in-
14 clude one or more of, or some combination of, the following applications:
15

- 16 • distribution of databases responses, including responses to common or fre-
17 quently used database queries; and
- 18 • distribution of email and groupware messages, bulletin board or newsgroup
19 messages.

20
21 Moreover, techniques used by a preferred embodiment of the invention for
22 content delivery can be used in contexts other than the specific applications disclosed

1 herein. For example, techniques used by embodiments of the invention for content deliv-
2 ery are all generally applicable to fields other than the specific applications disclosed
3 herein.

4

5 Other and further applications of the invention in its most general form
6 would be clear to those skilled in the art after perusal of this application. The invention
7 would be usable for such other and further applications without undue experimentation or
8 further invention.

9
10
11
12
13

Although preferred embodiments are disclosed herein, many variations are
possible which remain within the concept, scope and spirit of the invention; these varia-
tions would be clear to those skilled in the art after perusal of this application.